

**Macoun**

# CarPlay in Theorie & Praxis

Dr. Mickey Lauer  
Code & Words & Music

# Agenda

- Einordnung & Geschichte
- Technologie & APIs
- Audio
- Navigation
- Zusammenfassung & Ausblick

# Einordnung



„CarPlay ist eine von dem Computer- und Smartphone-Hersteller Apple entwickelte Technologie, um verschiedene Funktionen eines iPhones mit der Kommunikationsanlage in Kraftfahrzeugen zu nutzen.“ – Wikipedia



„CarPlay ist eine I/O-Erweiterung zur Integration von iPhone-Geräten in die Bordsysteme eines Automobils“ – Ich



# Geschichte

2003: iPod mit Multifunktionsbus (Dock Connector)

2010: iOS 4 mit „iPod Out“ (BMW & Apple)

2013: Ankündigung auf WWDC als „iOS in the Car“

2014: „CarPlay“ Launch in Genf (Ferrari, MB, Volvo) iPhone 5 + iOS 7.1

2015: iOS 9 mit Wireless CarPlay

2018: iOS 12 mit öffentlichem API für Navigations-Apps

2019: iOS 13 mit neuem Look, neuen Features und neuen Apps

# Technologie





# Technologie

(Auto-)Hardware

Tasten, Knöpfe, Mikrophone, Lautsprecher, Display, TouchScreen, Lenkrad

L0

# Technologie

USB

WiFi + Bluetooth

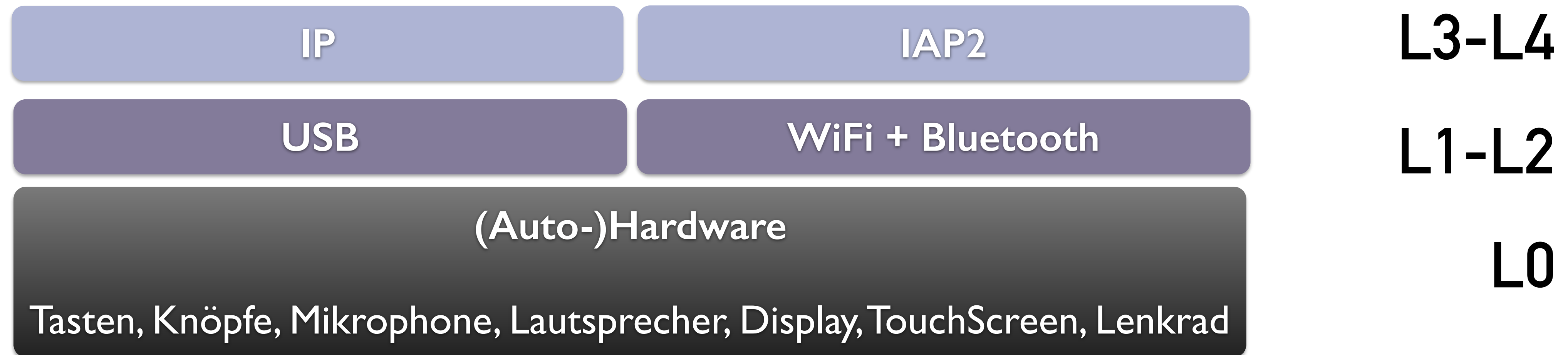
L1-L2

(Auto-)Hardware

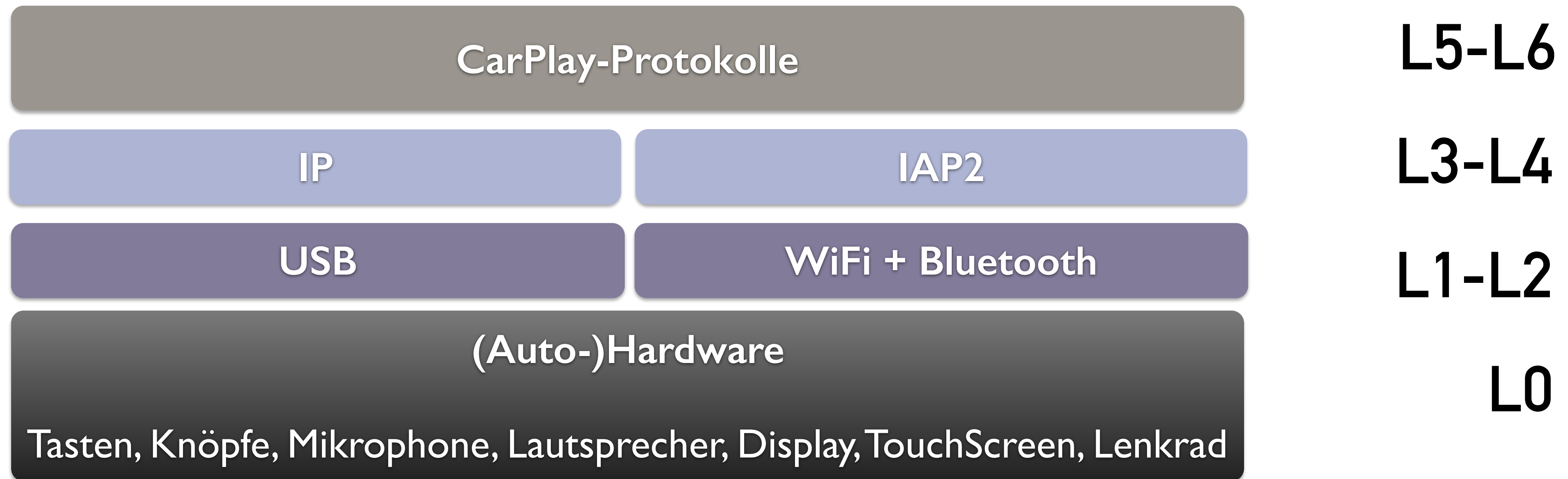
L0

Tasten, Knöpfe, Mikrophone, Lautsprecher, Display, TouchScreen, Lenkrad

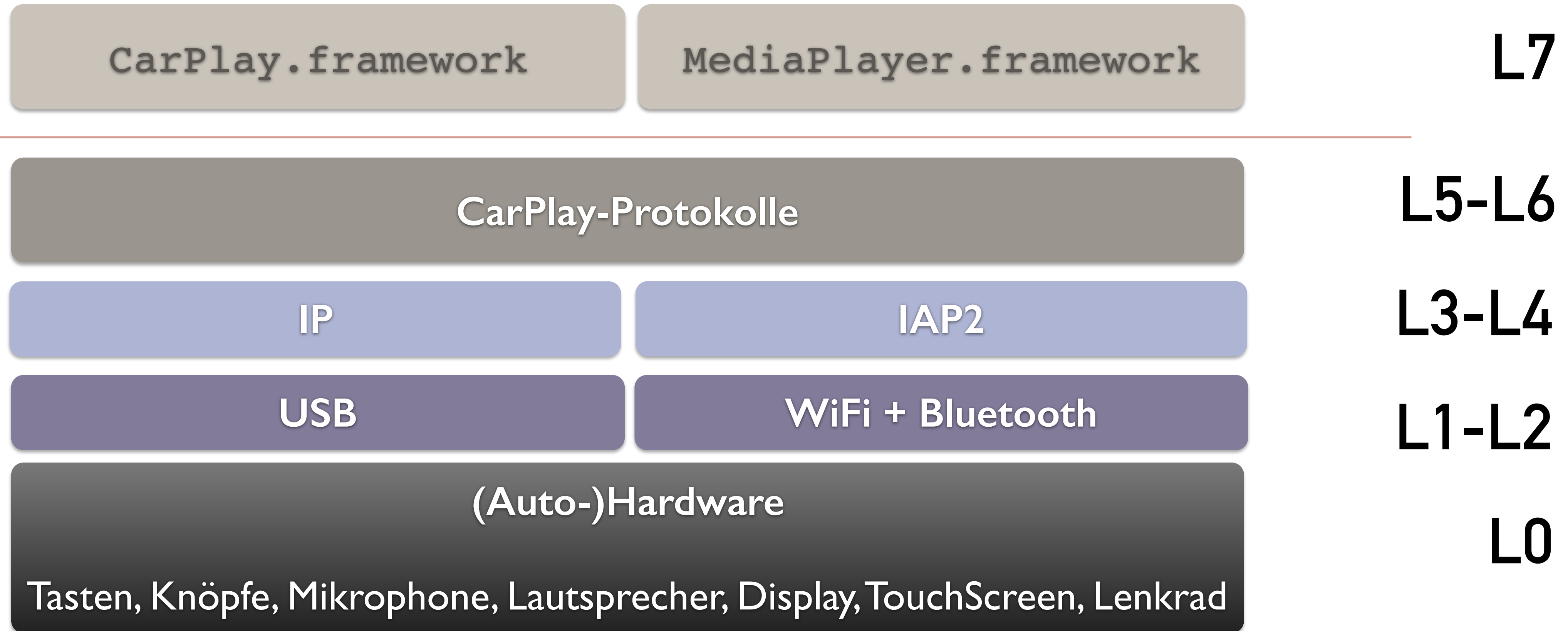
# Technologie



# Technologie



# Technologie




# CarPlay-Audio-App

1. Erscheinen auf dem CarPlay-Homescreen
2. Abschnitte und Listen zum *Durchbrowsen* veröffentlichen
3. Implementieren der Abspielkontrolle und „NowPlaying“

# CarPlay-Audio-App

Erscheinen auf dem  
CarPlay-Homescreen

# https://developer.apple.com/contact/carplay/

Developer Discover Design Develop Distribute Support Account 

Michael Lauer | [Sign Out](#)

## Thank you for your interest in CarPlay.

If you think your app has the potential to be supported by CarPlay, tell us about it.

Name Michael Lauer

Email drmickey@vanille.de

Organization Michael Lauer

App type  Select Option  
 Audio  
 Automaker  
 Messaging  
 Navigation  
 Other

Tell us about your product.





Von: carplay\_developer@apple.com  
Betreff: Re: [AUDIO] Requesting CarPlay  
Datum: 30. April 2018 um 21:36  
An: drmickey@vanille.de



Please include the line below in follow-up emails for this request.

Follow-up: 689665810

Dear Developer,

Thank you for your interest in developing a CarPlay audio app.

The following documents are attached:

- CarPlay Audio App Programming Guide
- CarPlay APIs Addendum to the iOS Developer Program License Agreement

Also, see Human Interface Guidelines for CarPlay Apps,  
<http://developer.apple.com/carplay/human-interface-guidelines/>

and the WWDC 2017 video Enabling Your App for CarPlay.  
<https://developer.apple.com/videos/play/wwdc2017/719/>

In order to receive the CarPlay audio app entitlement, please review, sign and return the Addendum by replying to this email.

We look forward to seeing your app on CarPlay.

Best regards,  
CarPlay Developer

On April 19, 2018 at 19:32:11 PM GMT, drmickey@vanille.de wrote:

Organization: Michael Lauer

Team ID: NANNL9SK66

Product:

Wellenreiter is an advanced internet streaming radio app with a hand-curated directory. See  
<https://itunes.apple.com/en/app/wellenreiter/id1069855738?mt=8>



CarPlay  
Addendum.pdf



CarPlay Audio  
App Pr...ide.pdf

# Entitlement hinzufügen

```
diff --git a/ios/Wellenreiter/Wellenreiter/Wellenreiter.entitlements b/ios/Wellenreiter/Wellenreiter/Wellenreiter.entitlements
index 0376977..67040db 100644
--- a/ios/Wellenreiter/Wellenreiter/Wellenreiter.entitlements
+++ b/ios/Wellenreiter/Wellenreiter/Wellenreiter.entitlements
@@ -2,6 +2,8 @@
 <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
 <plist version="1.0">
 <dict>
+   <key>com.apple.developer.playable-content</key>
+   <true/>
   <key>aps-environment</key>
   <string>development</string>
   <key>com.apple.developer.associated-domains</key>
```



10:35



3G



WhatsApp



GoogleMaps



Wellenreiter



MODE



1

2

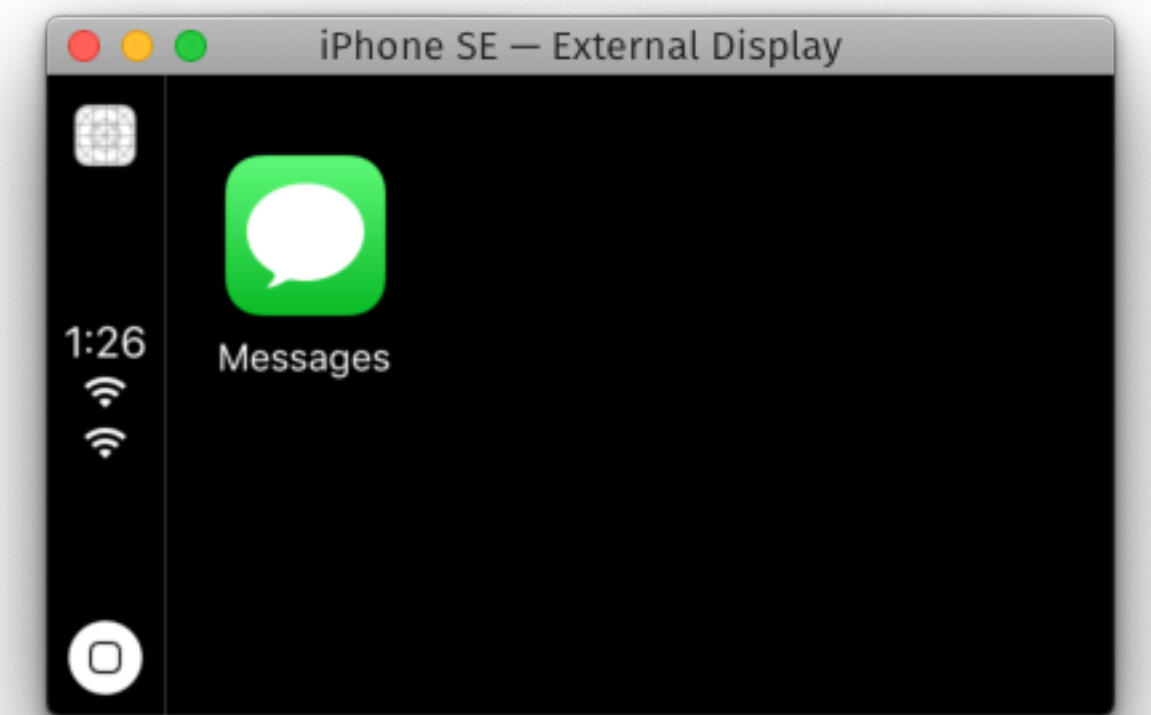
3

4

5

6

# Macht nix!?



MacounCarPlayAudioDemo > iPhone SE

MacounCarPlayAudioDemo

PROJECT: MacounCarPlayAudioDemo

TARGETS: MacounCarPlayAudioDemo

Basic | Customized | All | Combined | Levels | +

▼ Deployment

Setting	MacounCarPlayAudioDemo
Installation Directory	/Applications
Strip Linked Product	Yes
Targeted Device Family	Universal

▼ Linking

Setting	MacounCarPlayAudioDemo
Runpath Search Paths	@executable_path/Frameworks

▼ Packaging

Setting	MacounCarPlayAudioDemo
Info.plist File	MacounCarPlayAudioDemo/Info.plist
Product Bundle Identifier	de.vanille.MacounCarPlayAudioDemo
Product Name	MacounCarPlayAudioDemo

▼ Signing

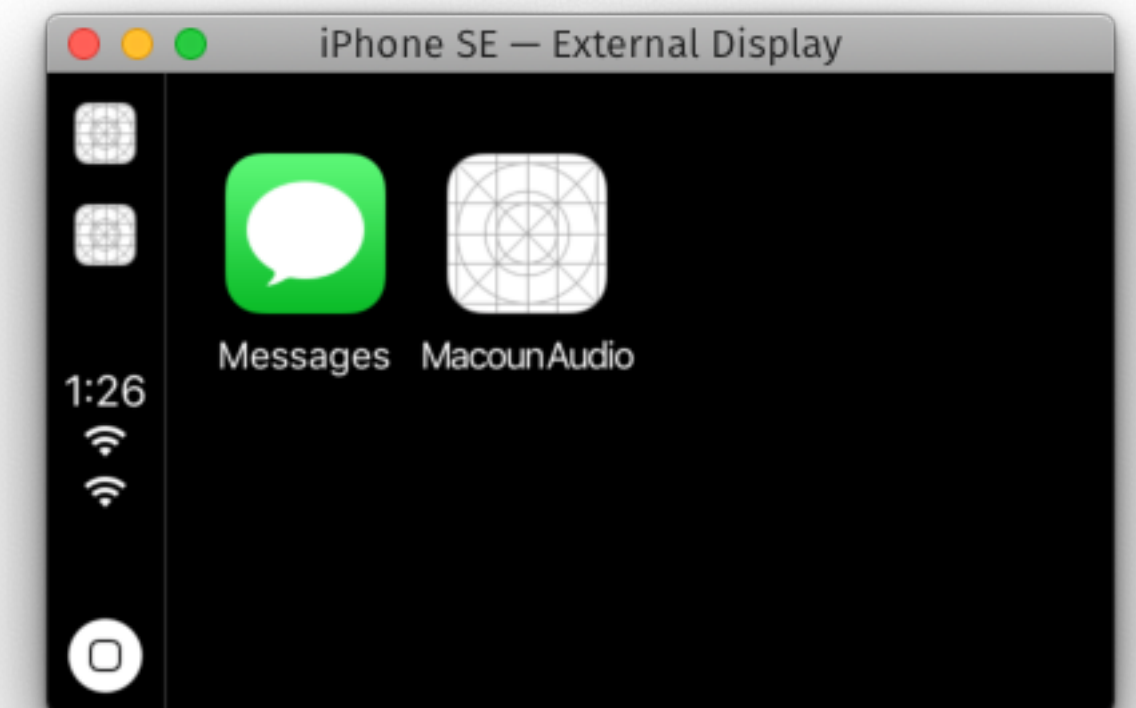
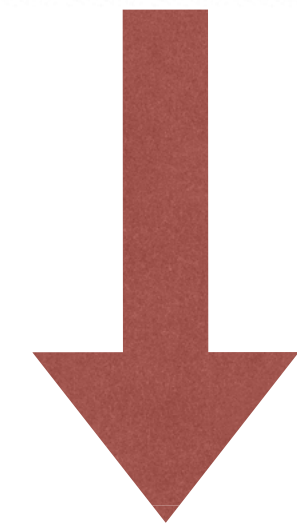
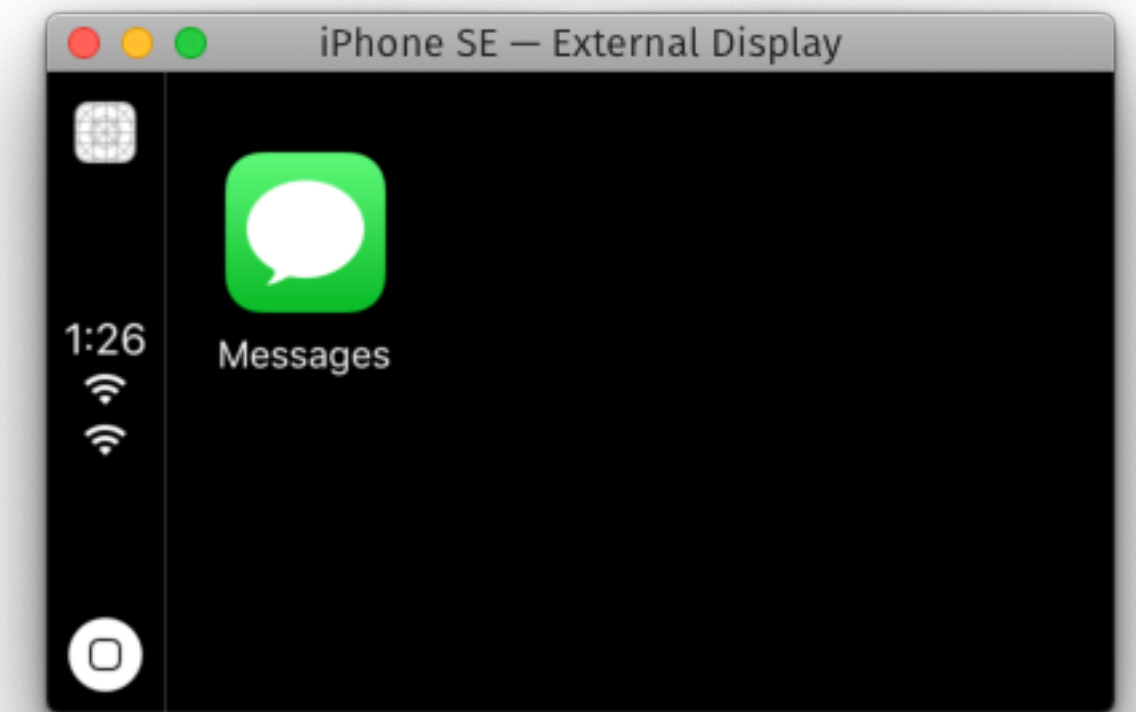
Setting	MacounCarPlayAudioDemo
Code Signing Entitlements	MacounCarPlayAudioDemo/MacounCarPlayAudioDemo.entitlements
Code Signing Style	Automatic
Development Team	Michael Lauer

▼ Asset Catalog Compiler - Options

Setting	MacounCarPlayAudioDemo
Asset Catalog App Icon Set Name	AppIcon

▼ User-Defined

Setting	MacounCarPlayAudioDemo
MTL_ENABLE_DEBUG_INFO	<Multiple values>
Debug	INCLUDE_SOURCE
Release	NO
MTL_FAST_MATH	YES



# CarPlay-Audio-App

Abschnitte und Listen zum  
Durchbrowsen veröffentlichen

CarPlay  
Application  
Window

CarPlay  
Audio  
App

**MPPlayableContentManager**

```
id<MPPlayableContentDataSource> dataSource  
id<MPPlayableContentDelegate> delegate;  
-(void)reloadData;
```



## MPContentItem

```
NSString* identifier;  
NSString* title;  
NSString* subtitle;  
MPMediaItemArtwork* artwork  
BOOL container;  
BOOL playable;  
BOOL streamingContent;  
float playbackProgress;  
BOOL explicitContent;
```

## MPMediaItemArtwork

```
-(instancetype)initWithBoundsSize:  
(CGSize)boundsSize requestHandler:(UIImage*  
_Nonnull (^)(CGSize size))requestHandler;
```

CarPlay  
Application  
Window

CarPlay  
Audio  
App

## MPPlayableContentManager

```
id<MPPlayableContentDataSource> dataSource  
id<MPPlayableContentDelegate> delegate;  
-(void)reloadData;
```

## <MPPlayableContentDataSource>

```
-(NSInteger)numberOfChildItemsAtIndexPath:(NSIndexPath *)indexPath;  
-(MPCContentItem *)contentItemAtIndexPath:(NSIndexPath *)indexPath;
```

## <MPPlayableContentDelegate>

```
-(void)playableContentManager:(MPPlayableContentManager *)contentManager  
initiatePlaybackOfContentItemAtIndexPath:(NSIndexPath *)indexPath  
completionHandler:(void (^)(NSError *))completionHandler;
```

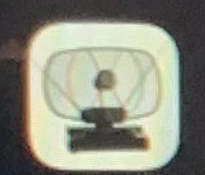
## <MPPlayableContentDataSource>

```
-(NSInteger)numberOfChildItemsAtIndexPath:(NSIndexPath *)indexPath;  
-(MPCContentItem *)contentItemAtIndexPath:(NSIndexPath *)indexPath;
```

## <MPPlayableContentDelegate>

```
-(void)playableContentManager:(MPPlayableContentManager *)contentManager  
initiatePlaybackOfContentItemAtIndexPath:(NSIndexPath *)indexPath  
completionHandler:(void (^)(NSError *))completionHandler;
```

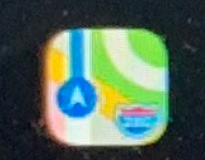
Achtung: NSIndexPath wird hier im *allgemeinen* Sinne der Adressierung von Elementen einer Baumstruktur benutzt!



< Wellenreiter Alle



Bremen NEXT  
Black, Urban und Hip Hop – Musik u...



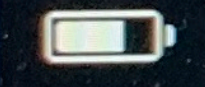
Bremen Vier  
Der coolste Radiosender der Welt, i...

12:19



ByteFM  
Hamburger Webradio für gute Musik

3G



CGM UK DemoScene Radio  
DemoScene Radio



COUNTRY 108



MODE



1

2

3

4

5

6

# CarPlay-Audio-App

Implementieren der  
Abspielkontrolle und  
„NowPlaying“



Wellenreiter

< Alle

# SomaFM DefCon

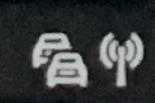
Guy J - Spica

13:25

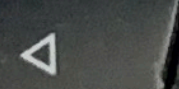
3G

⏪ ⏸ ⏩

⏏



MODE



1

2

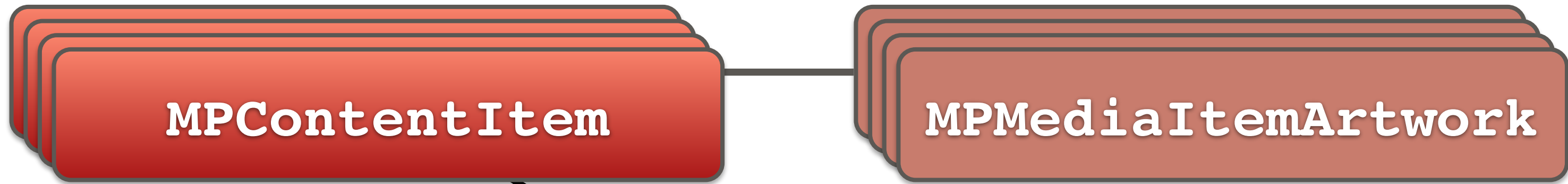
3

4

5

6





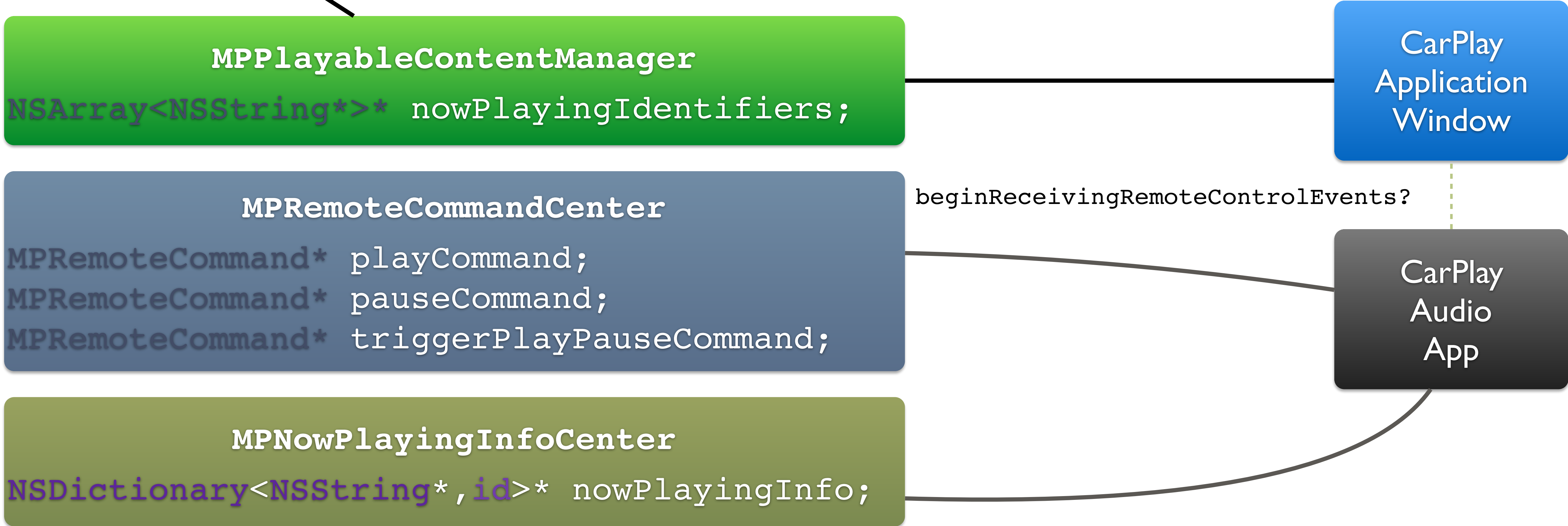
```
MPPlayableContentManager  
NSArray<NSString*>* nowPlayingIdentifiers;
```

```
MPRemoteCommandCenter  
MPRemoteCommand* playCommand;  
MPRemoteCommand* pauseCommand;  
MPRemoteCommand* triggerPlayPauseCommand;
```

```
MPNowPlayingInfoCenter  
NSDictionary<NSString*,id>* nowPlayingInfo;
```



beginReceivingRemoteControlEvents?





**Demo**

# CarPlay-Navigation-App

1. Erscheinen auf dem CarPlay-Homescreen
2. Interaktionslogik implementieren
3. Kartenansicht einbauen
4. Routenauswahl anzeigen
5. Navigationshinweise anzeigen

# CarPlay-Navigations-App

Erscheinen auf dem  
Home-Screen

# Entitlement hinzufügen

```
diff --git a/ios/StreetAssistant/StreetAssistant/StreetAssistant.entitlements
b/ios/StreetAssistant/StreetAssistant/StreetAssistant.entitlements
index 0376977..67040db 100644
--- a/ios/StreetAssistant/StreetAssistant/StreetAssistant.entitlements
+++ b/ios/StreetAssistant/StreetAssistant/StreetAssistant.entitlements
@@ -2,6 +2,8 @@
 <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/
DTDs/PropertyList-1.0.dtd">
 <plist version="1.0">
 <dict>
+   <key>com.apple.developer.carplay-maps</key>
+   <true/>
   <key>aps-environment</key>
   <string>development</string>
   <key>com.apple.developer.associated-domains</key>
```

# CarPlay-Navigations-App

Interaktionslogik  
implementieren

CarPlay  
Navigation  
App

**CPWindow : UIWindow**

**CPApplicationDelegate**

```
didConnectCarInterfaceController:toWindow:  
didDisconnectCarInterfaceController:fromWindow:
```

**CPInterfaceController**

```
CPTemplate* rootTemplate;  
pushTemplate:animated:  
popTemplateAnimated:  
presentTemplate:animated:  
dismissTemplateAnimated:
```

**CPTemplate**

```
id userInfo;
```

## UIViewController

```
UIView* view;
```

CarPlay  
Navigation  
App

## CPWindow : UIWindow

```
UIViewController* rootViewController;
```

## CPApplicationDelegate

```
didConnectCarInterfaceController:toWindow:  
didDisconnectCarInterfaceController:fromWindow:
```

## CPTemplate

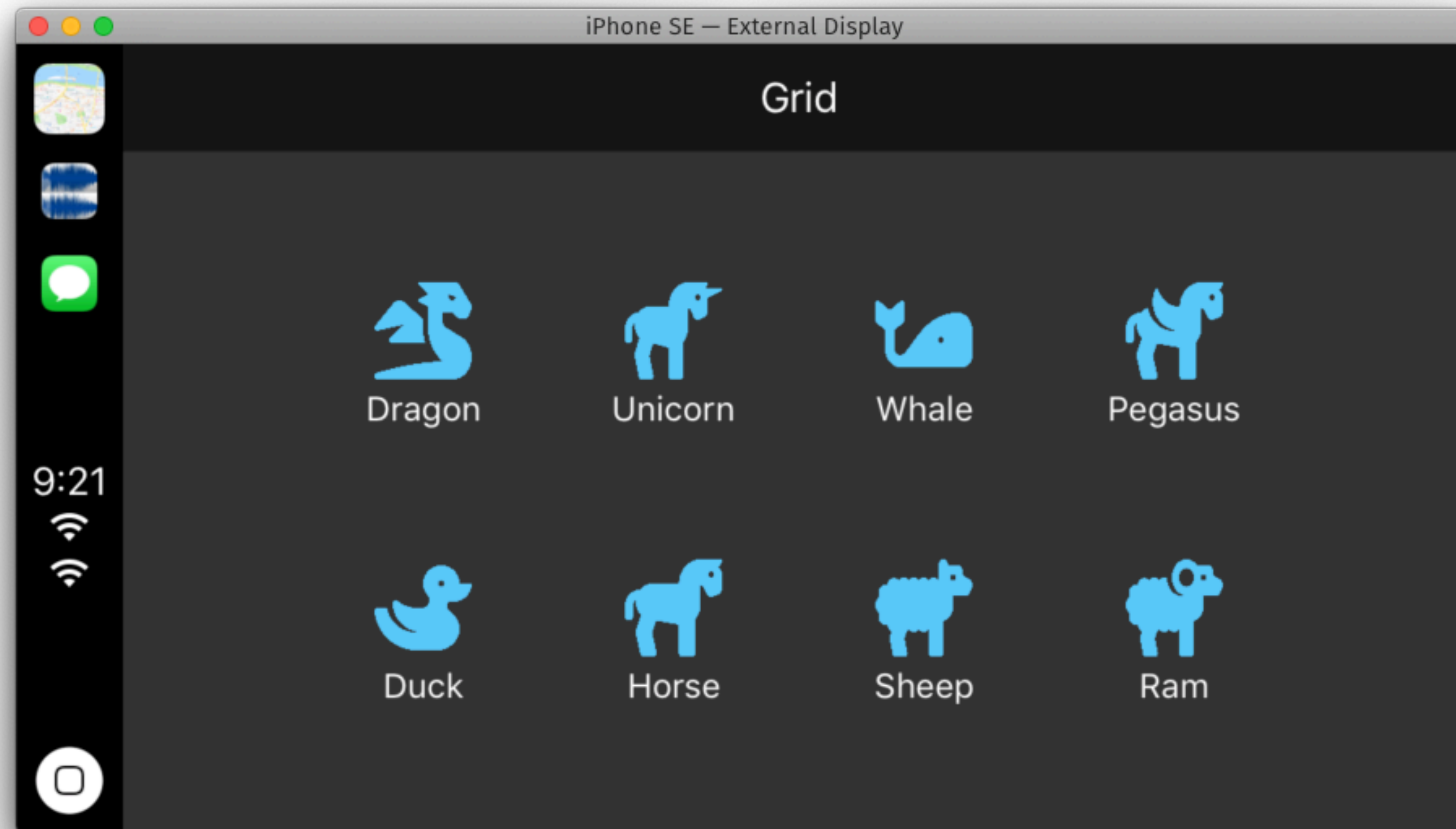
```
id userInfo;
```

## CPInterfaceController

```
CPTemplate* rootTemplate;  
pushTemplate:animated:  
popTemplateAnimated:  
presentTemplate:animated:  
dismissTemplateAnimated:
```

# CPTemplate

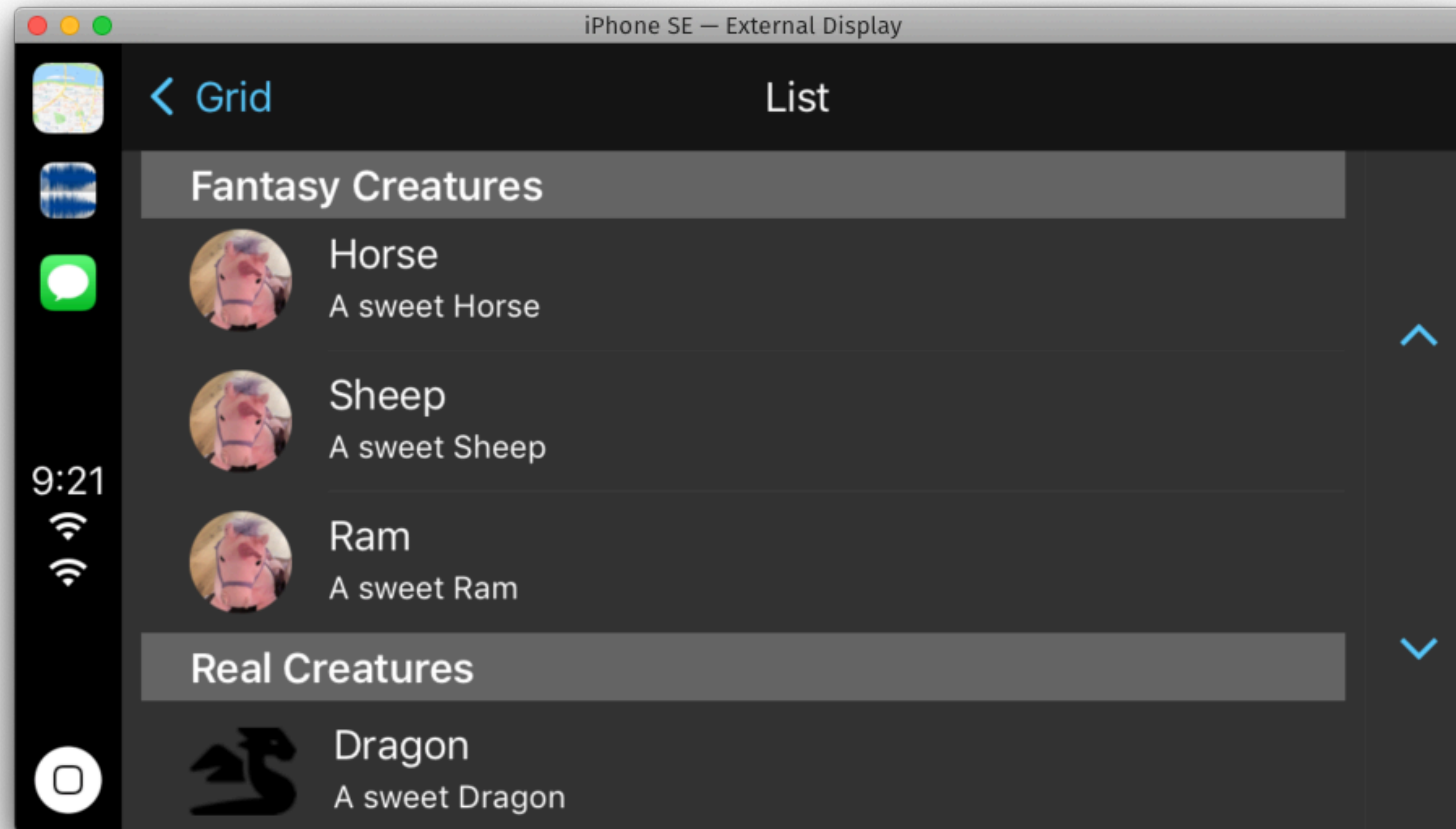
- **CPGridTemplate** Balanciertes Grid mit bis zu 8 Schaltflächen





# CPTemplate

- **CPListTemplate** Liste mit Elementen in Sektionen (vgl. UITableView)



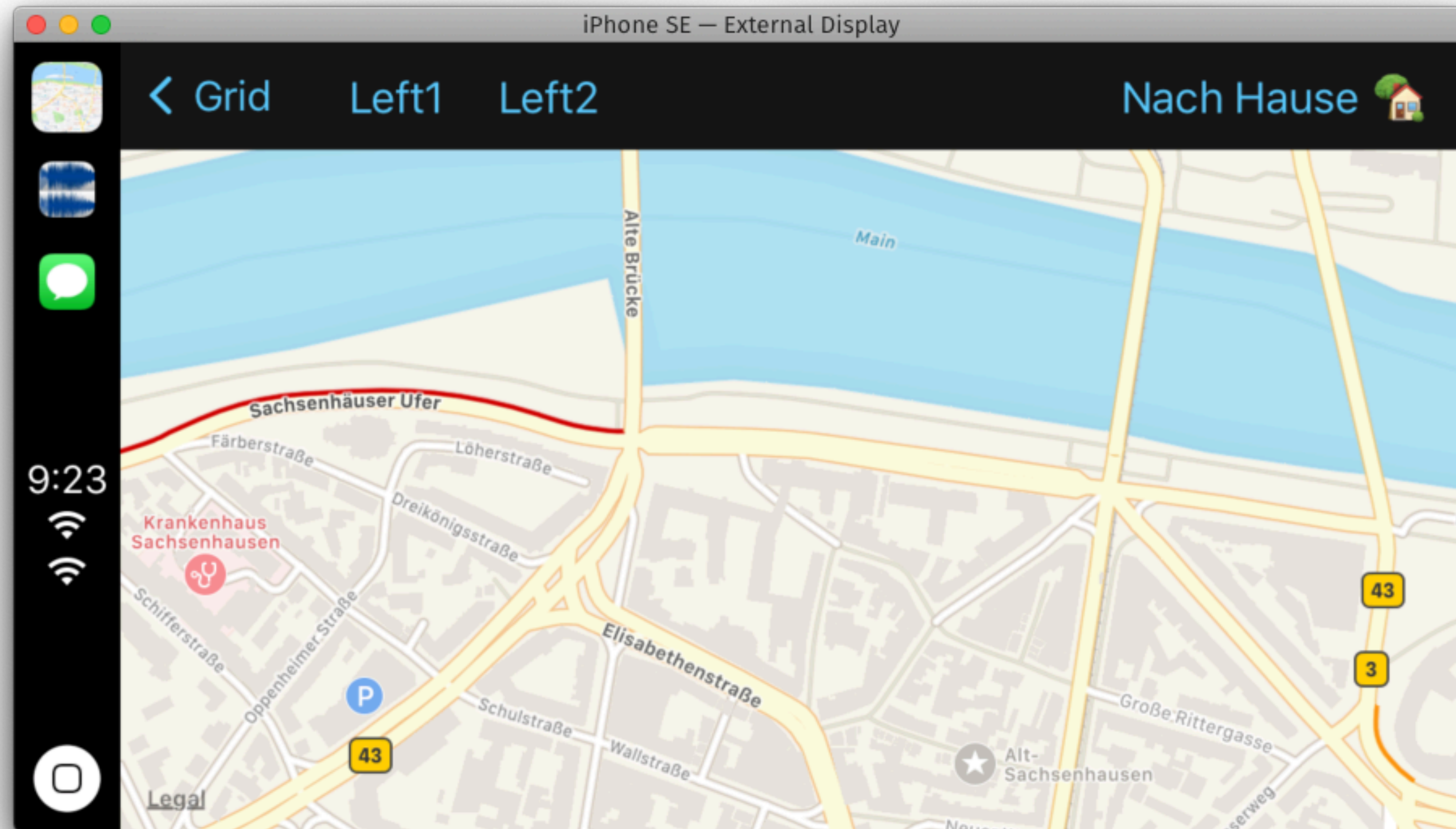
# CPTemplate

- CPSearchTemplate Suchfeld mit Tastatur und Ergebnisliste



# CPTemplate

- CPMaPTemplate Semitransparentes Overlay für Kartendarstellung



# CPTemplate (Nur Modal)

- **CPAlertTemplate** Pendant zu UIAlertController mit UIAlertControllerStyleAlert
- **CPActionSheetTemplate** Pendant zu UIAlertController mit UIAlertControllerStyleActionSheet
- **CPVoiceControlTemplate** Spracheingabe
- **Achtung: Template-Klassen sind „final“:**

```
Terminating app due to uncaught exception 'NSInvalidArgumentException', reason: 'Unsupported object <GridDemoTemplate: 0x600000e4cfc0> <identifier: 792C4785-AFA0-46ED-A22A-7963D0C1B478, userInfo: (null)> passed to setRootTemplate:animated:. Allowed classes: {(CPSearchTemplate, CPMAPTemplate, CPLISTTemplate, CPGridTemplate)}
```

# CPInterfaceController

- Kontrolliert Anzeige von CPTemplate-Instanzen
- Vom System erzeugt
- Lebenszyklus synchron mit der Verbindung zu CarPlay
- Enthält Stapelnavigation (vgl. UINavigationController)
  - `push`, `pop`, `popToRoot`, `popToTemplate`
- Modale Präsentation möglich – aber nicht verschachtelt
  - `present`, `dismiss`

# CarPlay-Navigations-App

Kartenansicht implementieren

# Kartenansicht

- Wie gewünscht im `UIViewController` mit `UIViews`
- `CMapTemplate` aktivieren, um Karte sichtbar zu machen
- Spezieller Modus für interaktive Bewegung der Karte möglich
  - Keine Events oder Gesten, nur spezielle Delegate-Methoden
- Theoretisch *vielen* darstellbar
- Praktisch entscheidet – wie immer – das Review
  - In-house via *Enterprise Distribution* möglich

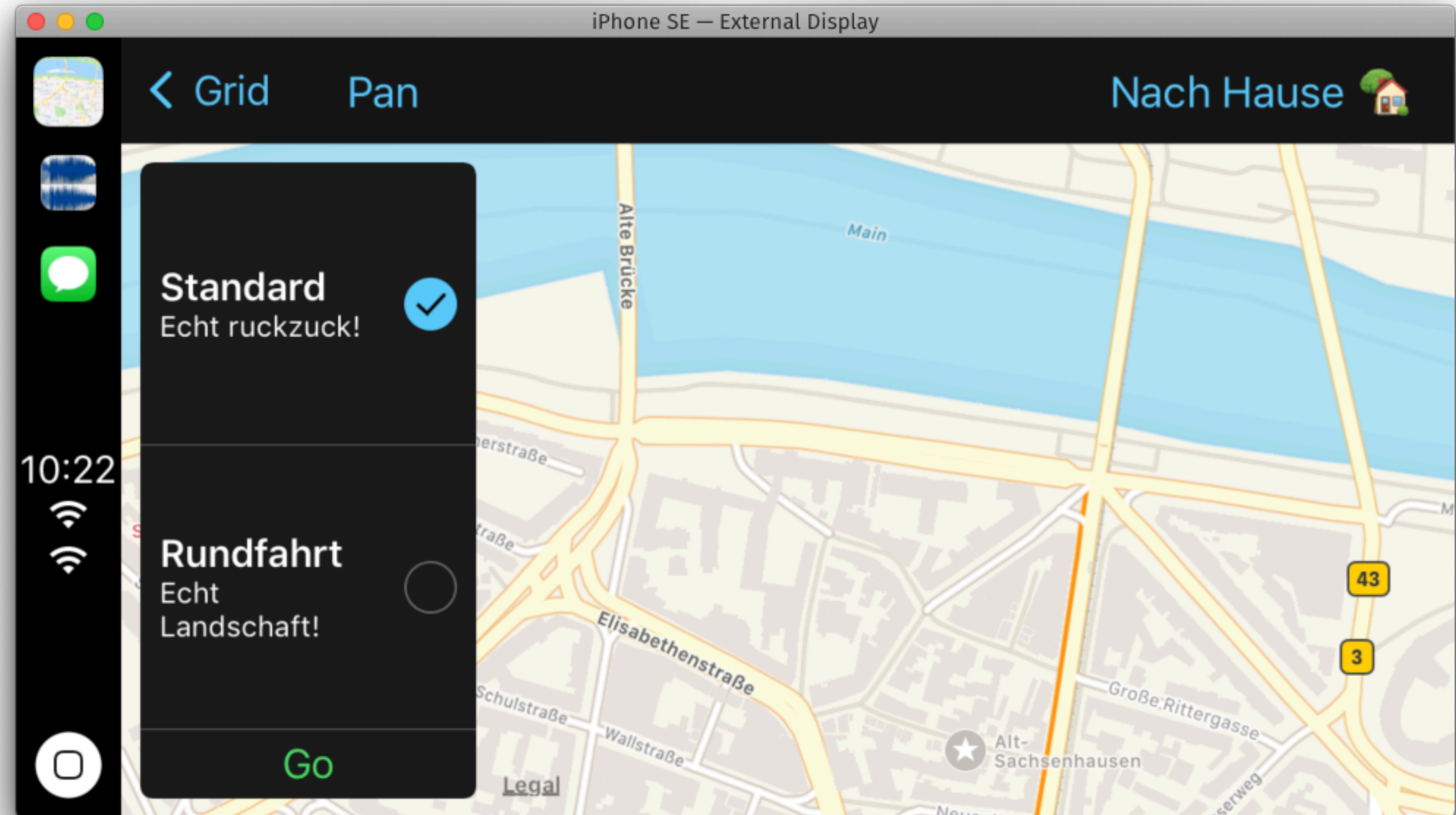
# CarPlay-Navigations-App

Routenauswahl zeigen



# Routenauswahl

- CPMRouteChoice erstellen
- CPTrip anlegen
- Routenvorschau anzeigen
- CPMapTemplateDelegate implementieren
- CPNavigationSession starten



# CarPlay-Navigations-App

Navigationshinweise zeigen

# Navigationshinweise

- CPManeuver anlegen
- upcomingManeuvers auf CPNavigationSession setzen
- CPTravelEstimates berechnen
- Aktualisieren via `updateTravelEstimates`  
`:trip:`



**Demo**

# Zusammenfassung & Ausblick



# Zusammenfassung

- Integration des iPhone in Automobil-Bordsysteme
- Öffentliches API derzeit limitiert auf Audio und Navigation
  - Keine UI-Interaktion bei Audio
  - UI-Interaktion bei Navigation nur über Proxy-Elemente
- Notwendige *Entitlements* vergibt Apple auf Anfrage
  - Ohne *Entitlements* ist nur Simulatorbetrieb möglich

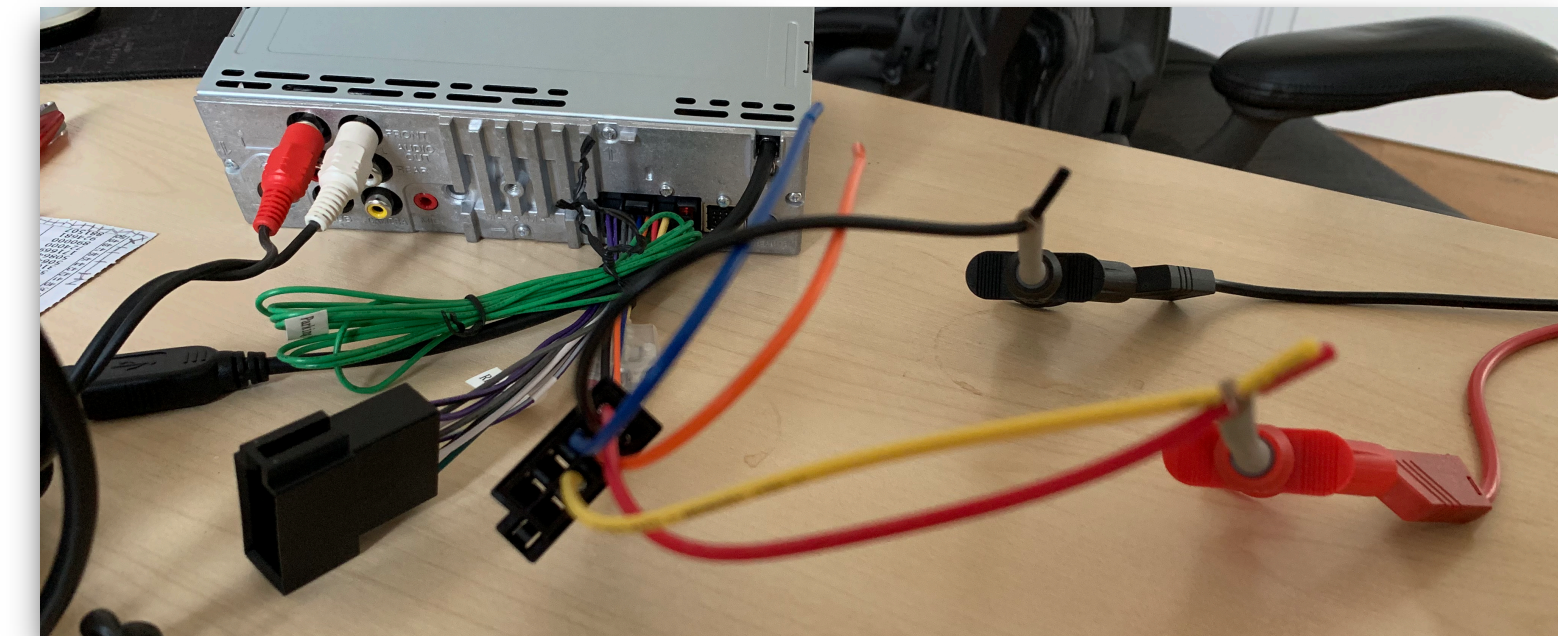
# Tipps

- Der CarPlay-Simulator ist mit (großer) Vorsicht zu genießen (Reboot tut gut)
- Konfiguration der „automobilen Hardware“ möglich via  
`defaults write com.apple.iphonesimulator CarPlayExtraOptions -bool YES`
- Wired CarPlay und Wireless CarPlay verhalten sich unterschiedlich
- Testen im Auto (besser: in mehreren) unerlässlich
- Große Displayfragmentierung: `NSArray<NSString*>*` Textvarianten **nutzen!**
- Man kann ein eigenes Icon für die CarPlay-App anlegen
- Zu große Bilder werden (meistens) einfach ignoriert

# Setup



- Autoradio, z.B. Sony XAV-AX100
- Labornetzteil, z.B. IPSXP KPS 1203D
- ISO-Adapter auf blanke Litzen, z.B. Audioproject AI 54
- Aktivboxen oder z.B. Habit Bluetooth Lautsprecher Box
- Kleinteilige Adapter und Kabel  
(Chinch auf Miniklinke, Krokodilklemmen, Bananenstecker, etc.)





# Quellen

- <https://www.vanille.de/data/carplay.pdf>
- <https://github.com/mickey1/iOS-Bug-Example-Projects>
- <https://developer.apple.com/design/human-interface-guidelines/carplay/overview/audio-apps/>
- <https://developer.apple.com/carplay/documentation/CarPlay-Navigation-App-Programming-Guide.pdf>
- <https://stackoverflow.com/questions/tagged/carplay?tab=Newest>

# Ausblick

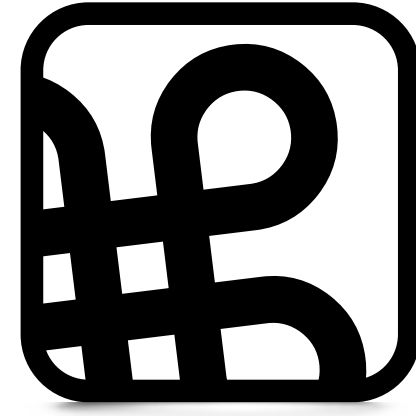
- Was macht „die Konkurrenz“? Nicht viel anderes.
  - Android Auto
  - Automotive Linux et. al.
- Situation vergleichbar mit Mobilfunk Anfang der 2000er
  - Derzeit hauptsächlich noch „In-Vehicle-Infotainment“
  - Sinnvolle Anwendungen außerhalb dieser Kategorie?
- Phase des automobilen „Umbruchs“

*Danke für die Aufmerksamkeit!*

# **CarPlay in Theorie & Praxis**

<https://www.vanille.de>

@DrMickeyLauer



**Macoun**